

案例 1-基本图元绘制算法

文档编写：霍波魏

校稿/修订：孔令德

时间 2019~2020

联系方式：QQ997796978

说明：本套案例由孔令德开发，原版本为 Visual C++6.0，配套于孔令德的著作《计算机图形学-基于 MFC 三维图形开发》一书。孔令德计算机工程研究所的学生霍波魏在学习计算机图形学期间，对本套案例进行了升级并编写了学习文档。现在程序的编写和程序的解释都是基于 Windows 10 操作系统，使用 Microsoft visual studio 2017 平台的 MFC（英文版）开发。

一、案例描述

基本图元绘制算法较为简单，主要是目的是通过鼠标实现图形的绘制，我们以直线、矩形和椭圆三个基本图形为例进行程序的编写，并实现在状态栏对鼠标坐标的实时显示功能。

二、实现步骤

- 1、在 onDraw 函数中通过 MoveTo(),LineTo()函数绘制图形
- 2、添加 OnLButtonDown 函数和 OnLButtonUp 函数，实现点击鼠标开始移动，松开鼠标绘制图形的目的
- 3、添加 OnMouseMove()函数,实现在状态栏上对鼠标坐标的实时显示功能

三、主要算法

1. CTestView 类

```
void CTestView::OnDraw(CDC* pDC)
{
    CTestDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;
    // TODO: 在此处为本机数据添加绘制代码
    //绘制直线
    pDC->MoveTo(p0.x, p0.y);
    pDC->LineTo(p1.x, p1.y);
    //绘制矩形
    //pDC->Rectangle(p0.x, p0.y, p1.x, p1.y);
    //绘制椭圆
```

```

        //pDC->Ellipse(p0.x, p0.y, p1.x, p1.y);
    }
// CTestView 消息处理程序
void CTestView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    p1 = point;
    Invalidate(FALSE);
    CView::OnLButtonUp(nFlags, point);
}
void CTestView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    p0 = point;
    CView::OnLButtonDown(nFlags, point);
}
void CTestView::OnDrawpic()
{
    // TODO: 在此添加命令处理程序代码
    MessageBox(_T("请按下鼠标左键绘图!"), _T("提示"), MB_ICONEXCLAMATION);
    RedrawWindow();
}
void CTestView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    CString strx, stry;           //状态栏显示鼠标位置
    CMainFrame* pFrame = (CMainFrame*)AfxGetApp()->m_pMainWnd;//要求包含 MainFrame.h
    头文件
    CStatusBar* pStatus = &pFrame->m_wndStatusBar;//需要将 m_wndStatusBar 属性修改为
    公有
    if (pStatus)
    {
        strx.Format(_T("x=%d"), point.x);
        stry.Format(_T("y=%d "), point.y);
        CDC* pDC = GetDC();
        CSize sizex = pDC->GetTextExtent(strx);
        CSize sizey = pDC->GetTextExtent(stry);
        pStatus->SetPaneInfo(1, ID_INDICATOR_X, SBPS_NORMAL, sizex.cx);//改变状态栏
        风格
        pStatus->SetPaneText(1, strx);
        pStatus->SetPaneInfo(2, ID_INDICATOR_Y, SBPS_NORMAL, sizey.cx);//改变状态栏
        风格
        pStatus->SetPaneText(2, stry);
        ReleaseDC(pDC);
    }
}

```

```
}  
CView::OnMouseMove(nFlags, point);  
}
```

四、实现效果

基本图元绘制效果如图 1-1、1-2、1-3、1-4 所示。

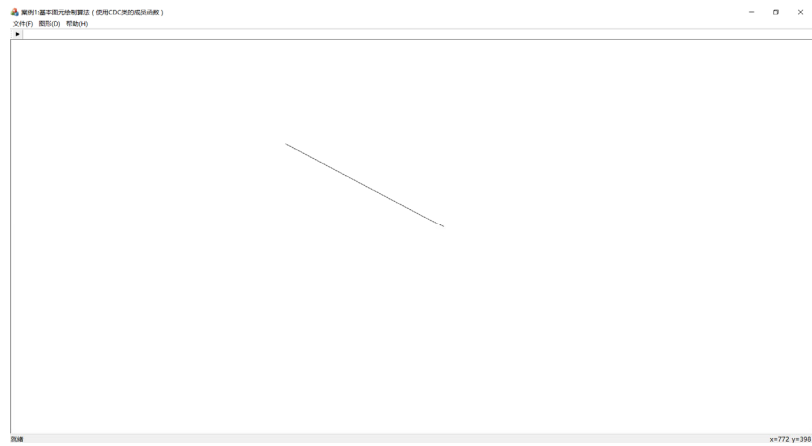


图 1-1 直线绘制效果图

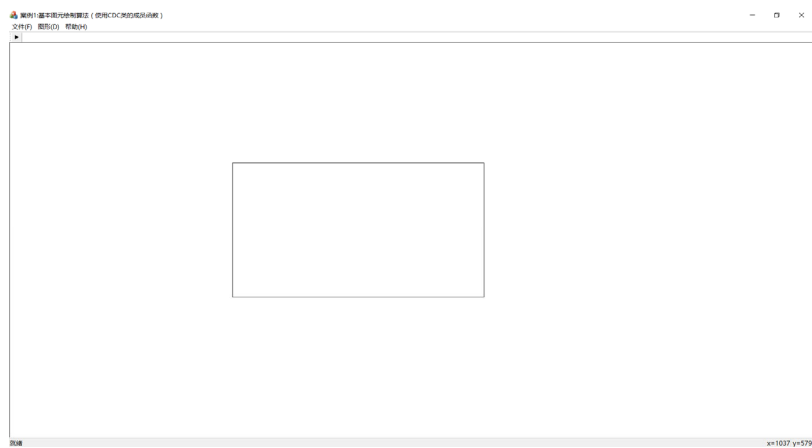


图 1-2 矩形绘制效果图

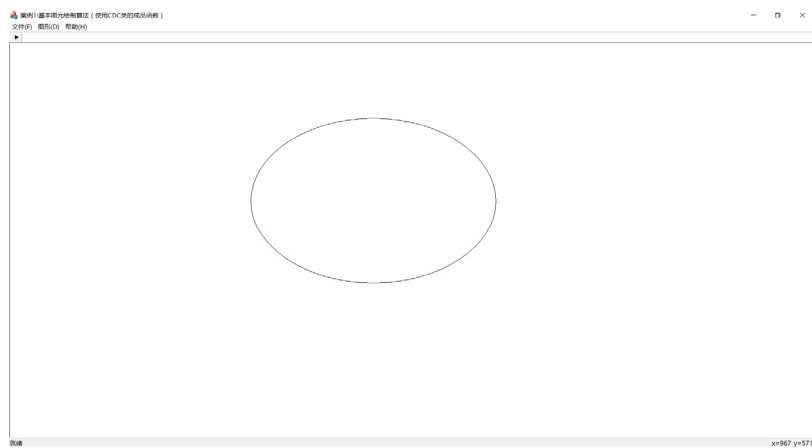


图 1-3 椭圆绘制效果图

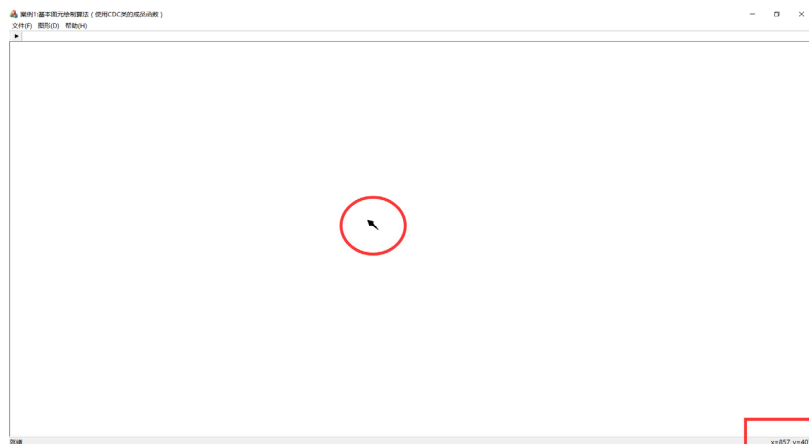


图 1-4 基本图元坐标显示效果图

五、遇到的问题及解决方案

1、从 VC6.0 到 VS2017 版本升级时，需要解决参数类型不匹配的问题，解决方法见图 1-5。

```
strx.Format("x=%d", point.x);
stry.Format("y=%d", point.y);
```

→

```
strx.Format(_T("x=%d"), point.x);
stry.Format(_T("y=%d"), point.y);
```

图 1-5 参数类型匹配

2、不会在状态栏上添加窗格，解决方案如下：

在状态栏资源中添加一个窗格。实现的方法是打开 MainFrm.cpp 文件，找到静态数组 indicators 的定义，在第一个数组元素 ID_SEPARATOR 后面增加一个新的数组元素，即添加了一个新的窗格，并对窗格进行命名，如图 1-6。

Test.rc - String Table		
ID	值	标题
IDP_OLE_INIT_FAILED	100	OLE 初始化失败。 请确保 OLE 库是正确的版本。
ID_INDICATOR_X	101	X
ID_INDICATOR_Y	102	Y

```
static UINT indicators[] =
{
    ID_SEPARATOR,           // 状态行指示器
    ID_INDICATOR_X,
    ID_INDICATOR_Y,
};
```

图 1-6 在状态栏上添加窗格

六、案例心得

巩固了 MoveTo、LineTo 函数的使用及简单图形的绘制，学会了如何在状态栏上实时显示鼠标坐标，也学会了如何在状态栏上添加窗格，同时也了解了鼠标消息响应函数。